Pattern Recognition

Bertrand Thirion and John Ashburner

1. Introduction
   - Classification and Regression
   - Curse of Dimensionality
2. Generalization of learned models across datasets
3. Overview of the main methods
4. Model Averaging

1. Introduction
   - Classification and Regression
   - Curse of Dimensionality
2. Generalization of learned models across datasets
3. Overview of the main methods
4. Model Averaging

General setting

We have a training dataset of n observations, each consisting of an input $\mathbf{x}_i$ and a target y_i.
Each input, $\mathbf{x}_i$, consists of a vector of p features.

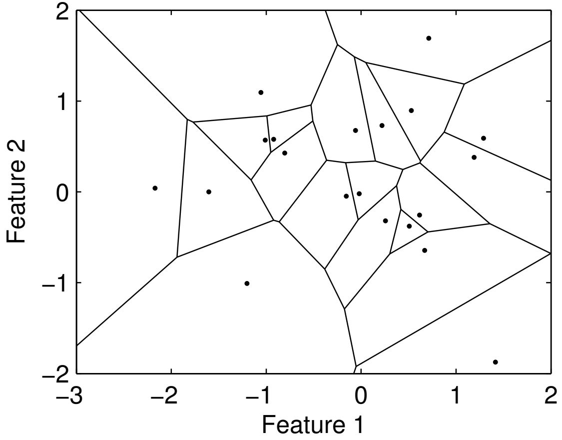
$$\mathcal{D} = \{(\mathbf{x}_i, y_i) | i = 1, .., n\}$$

The aim is to predict the target for a new input $\mathbf{x}_*$.

Curse of dimensionality

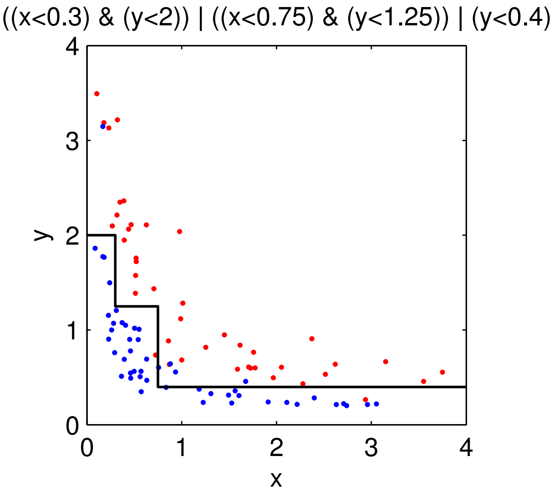
Large p, small n.

Nearest-neighbour classification



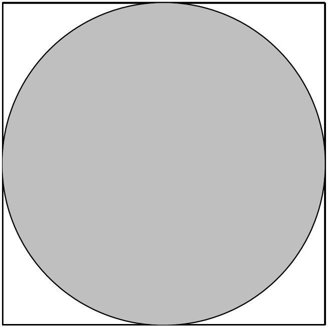
- Not nice smooth separations.
- Lots of sharp corners.
- May be improved with *K-nearest neighbours*.

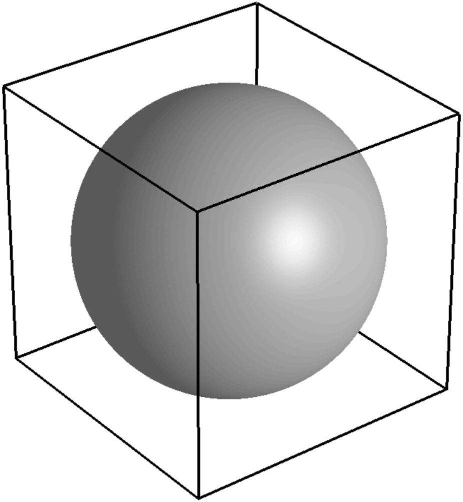
Rule-based approaches



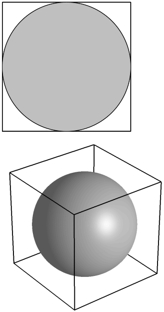
- Not nice smooth separations.
- Lots of sharp corners.

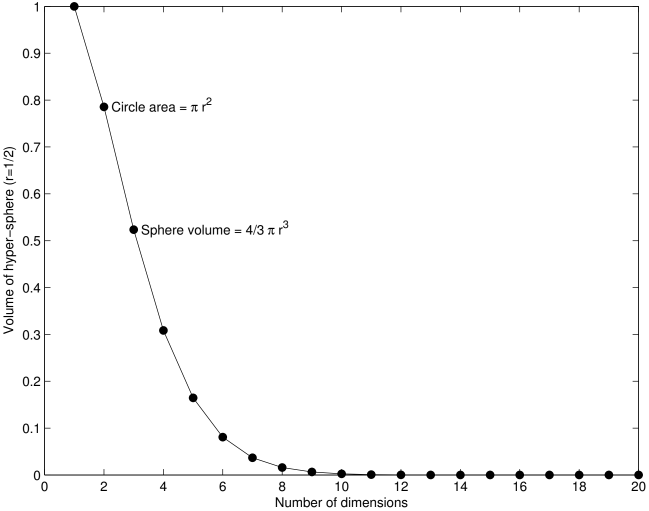
Corners matter in high-dimensions





Corners matter in high-dimensions





Occam's razor

"Everything should be kept as simple as possible, but no simpler."

— Einstein (allegedly)

- Complex models (with many estimated parameters) usually explain training data better than simpler models.
- Simpler models often generalise better to new data than nore complex models.

Need to find the model with the optimal bias/variance tradeoff.

Bayesian model selection

Real Bayesians don't cross-validate (except when they need to).

$$P(M|\mathcal{D}) = \frac{p(\mathcal{D}|M)P(M)}{P(\mathcal{D})}$$

The *Bayes factor* allows the plausibility of two models (M_1 and M_2) to be compared:

$$K = \frac{p(\mathcal{D}|M_1)}{p(\mathcal{D}|M_2)} = \frac{\int_{\theta_{M_1}} p(\mathcal{D}|\theta_{M_1}, M_1)p(\theta_{M_1}|M_1)d\theta_{M_1}}{\int_{\theta_{M_2}} p(\mathcal{D}|\theta_{M_2}, M_2)p(\theta_{M_2}|M_2)d\theta_{M_2}}$$

This is usually too costly in practice, so approximations are used.

Model selection

Some approximations/alternatives to the Bayesian approach:

- **Laplace approximations**: find the MAP/ML solution and use a Gaussian approximation to the parameter uncertainty.
- **Minimum Message Length** (MML): an information theoretic approach.
- **Minimum Description Length** (MDL): an information theoretic approach based on how well the model compresses the data.
- **Akaike Information Criterion** (AIC): $-2 \log p(\mathcal{D}|\theta) + 2k$, where k is the number of estimated parameters.
- **Bayesian Information Criterion** (BIC): $-2 \log p(\mathcal{D}|\theta) + k \log q$, where q is the number of observations.

Log predictive probability

Some data are more easily classified than others.
Probabilistic classifiers provide a level of confidence for each prediction.

$$p(y_*|\mathbf{x}_*, \mathbf{y}, \mathbf{X}, \theta)$$

Quality of predictions can be assessed using the **test log predictive probability**:

$$\frac{1}{m}\sum_{i=1}^{m} \log_2 p(y_{*i} = t_i|\mathbf{x}_{*i}, \mathbf{y}, \mathbf{X}, \theta)$$

After subtracting the baseline measure, this shows the average bits of information given by the model.

Rasmussen & Williams. "Gaussian Processes for Machine Learning", MIT Press (2006).
http://www.gaussianprocess.org/gpml/

1. Introduction
2. **Generalization of learned models across datasets**
   - Cross-Validation
   - Accuracy Measures
   - **Parameter Tuning**
3. Overview of the main methods
4. Model Averaging

1 Introduction

2 Generalization of learned models across datasets

3 Overview of the main methods
- Simple Methods: Naive Bayes, Linear Discriminant Analysis
- Kernel Methods: Support-Vector Machines, Gaussian Processes
- Basic Regularization Methods

4 Model Averaging

1. Introduction

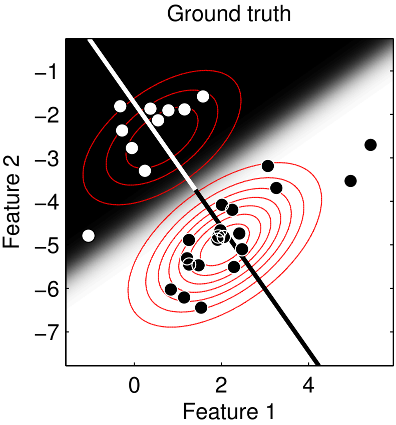
2. Generalization of learned models across datasets

3. Overview of the main methods
   - Simple Methods: Naive Bayes, Linear Discriminant Analysis
   - Kernel Methods: Support-Vector Machines, Gaussian Processes
   - Basic Regularization Methods

4. Model Averaging

Generative models for classification

$$P(y=k|\mathbf{x}) = \frac{P(y=k)p(\mathbf{x}|y=k)}{\sum_j P(y=j)p(\mathbf{x}|y=j)}$$

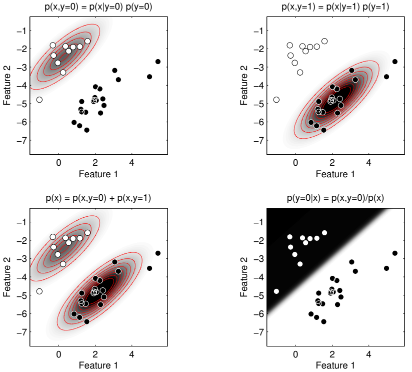


Linear discriminant analysis

$$P(y=k|\mathbf{x}) = \frac{P(y=k)p(\mathbf{x}|y=k)}{\sum_j P(y=j)p(\mathbf{x}|y=j)}$$

Assumes:

$$P(\mathbf{x}|y=k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$$



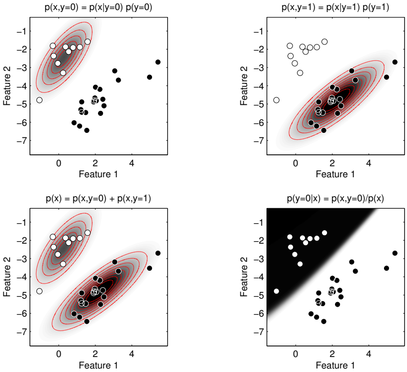
Model has $2p + p(p-1)$ parameters to estimate (two means and a single covariance).
Number of observations is pn (size of inputs).

Quadratic discriminant analysis

$$P(y=k|\mathbf{x}) = \frac{P(y=k)p(\mathbf{x}|y=k)}{\sum_j P(y=j)p(\mathbf{x}|y=j)}$$

Assumes different covariances:

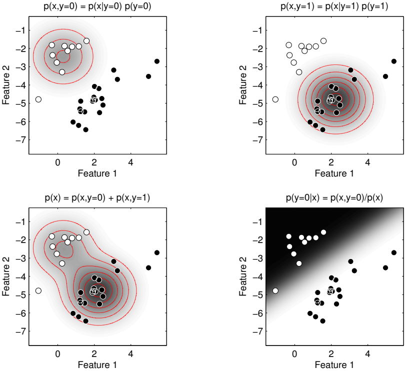
$$P(\mathbf{x}|y=k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$



Model has $2p + 2p(p-1)$ parameters to estimate (two means and two covariances).
Number of observations is pn.

Naive Bayes

$$P(y=k|\mathbf{x}) = \frac{P(y=k)p(\mathbf{x}|y=k)}{\sum_j P(y=j)p(\mathbf{x}|y=j)}$$



Assumes that features are independent:

$$p(\mathbf{x}|y=k) = \prod_i p(x_i|y=k)$$

Model has variable number of parameters to estimate, but the above example has $3p$.
Number of observations is pn.

Linear regression: maximum likelihood

$$f(\mathbf{x}_*) = \mathbf{a}^T \mathbf{x}_*$$

Assuming Gaussian noise on $\mathbf{y}$, the ML estimate of $\mathbf{a}$ is by:

$$\hat{\mathbf{a}} = (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{y}$$

where

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_n \end{pmatrix}^T, \text{ and } \mathbf{y} = \begin{pmatrix} y_1 & y_2 & \dots y_n \end{pmatrix}^T$$

Model has p parameters to estimate.
Number of observations is n (number of targets).

Linear regression: maximum posterior

$$y \sim \mathcal{N}(\mathbf{a}^T \mathbf{x}, \sigma^2)$$
$$\mathbf{a} \sim \mathcal{N}(\mathbf{0}, \Sigma_0)$$

Maximum a posteriori (MAP) estimate of **a** is by:

$$\hat{\mathbf{a}} = \sigma^{-2}\mathbf{C}^{-1}\mathbf{X}\mathbf{y}, \text{ where } \mathbf{C} = \sigma^{-2}\mathbf{X}\mathbf{X}^T + \Sigma_0^{-1}$$

Number of estimated parameters and observations is ill defined.

Linear regression: Bayesian

$$p(y_*|\mathbf{x}_*, \mathbf{a}) = \mathcal{N}(\mathbf{a}^T\mathbf{x}_*, \sigma^2)$$
$$p(\mathbf{a}|\mathbf{y}, \mathbf{X}) = \mathcal{N}(\sigma^{-2}\mathbf{C}^{-1}\mathbf{X}\mathbf{y}, \mathbf{C}^{-1}), \text{ where } \mathbf{C} = \sigma^{-2}\mathbf{X}\mathbf{X}^T + \Sigma_0^{-1}$$

$$\begin{aligned} p(y_*|\mathbf{x}_*, \mathbf{y}, \mathbf{X}) &= \int_{\mathbf{a}} p(y_*|\mathbf{x}_*, \mathbf{a}) p(\mathbf{a}|\mathbf{y}, \mathbf{X}) d\mathbf{a} \\ &= \mathcal{N}(\sigma^{-2}\mathbf{x}_*^T\mathbf{C}^{-1}\mathbf{X}\mathbf{y}, \mathbf{x}_*^T\mathbf{C}^{-1}\mathbf{x}_*) \end{aligned}$$

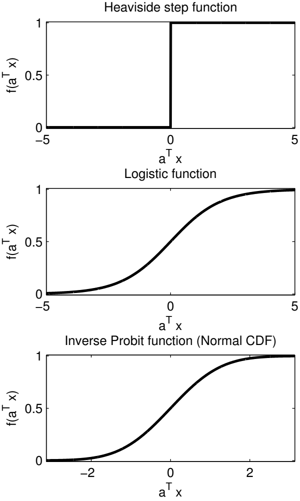
Weights are integrated out - rather than estimated.
Estimated parameters may be σ^2, and parameters encoding Σ_0.

Discriminative models for classification

$$t = f(\mathbf{a}^T \mathbf{x})$$

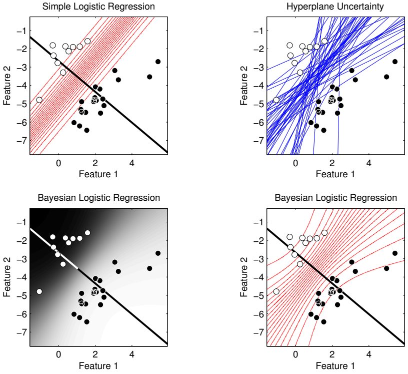
where f is some squashing function, eg:

- Heaviside step function.
- Logistic function (inverse of Logit).
- Normal CDF (inverse of Probit).



Probabilistic classification

$$P(y=k|\mathbf{x}) = \int_{\mathbf{a}} P(y=k|\mathbf{x},\mathbf{a})p(\mathbf{a})d\mathbf{a}$$



Woodbury matrix identity

$$\mathbf{C}^{-1} = \left(\sigma^{-2}\mathbf{X}\mathbf{X}^{T} + \Sigma_0^{-1}\right)^{-1}$$
$$= \Sigma_0 - \Sigma_0\mathbf{X}(\mathbf{I}\sigma^2 + \mathbf{X}^{T}\Sigma_0\mathbf{X})^{-1}\mathbf{X}\Sigma_0$$

Wikipedia contributors, "Woodbury matrix identity," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=Woodbury_matrix_identity&oldid=638370219 (accessed April 1, 2015).

Support vector classification

Targets are $\mathbf{t} \in \{-1, 1\}$.
Solves a quadratic programming problem

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \tfrac{1}{2}\boldsymbol{\alpha}^T \mathbf{H}\boldsymbol{\alpha} - \sum_{i=1}^{n} \alpha_i,$$

subject to $\mathbf{t}^T\boldsymbol{\alpha} = 0$ and $0 \leq \alpha_i \leq C$

$$\text{where } \mathbf{H} = \text{diag}(\mathbf{t})\mathbf{X}\mathbf{X}^T \text{diag}(\mathbf{t})$$

blah

Binary prediction is by:

$$t_* = sgn(\sum_{i=1}^{N} t_i \alpha_i \mathbf{x}_i \mathbf{x}_*^T + b)$$

where b is a bias term.

1. Introduction
2. Generalization of learned models across datasets
3. Overview of the main methods
4. Model Averaging
   - Decision trees and Random Forests
   - Boosting & Bagging
   - Tools: scikit-learn, pronto, nilearn, pymvpa

Ensemble learning

Combining predictions from weak learners.

- **Bootstrap aggregating (bagging)**
  - Train several weak classifiers, with different models or randomly drawn subsets of the data.
  - Average their predictions with equal weight.
- **Boosting**
  - A family of approaches, where models are weighted according to their accuracy.
  - AdaBoost is popular, but has problems with target noise.
- **Bayesian model averaging**
  - Really a model selection method.
  - Relatively ineffective for combining models.
- **Bayesian model combination**
  - Shows promise.

Monteith, et al. "Turning Bayesian model averaging into Bayesian model combination." Neural Networks (IJCNN), The 2011 International Joint Conference on. IEEE, 2011.